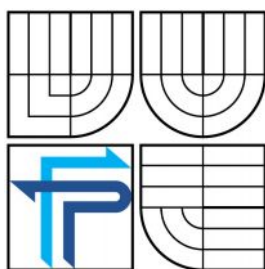


**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA PODNIKATELSKÁ**  
**ÚSTAV INFORMATIKY**

FACULTY OF BUSINESS AND MANAGEMENT  
INSTITUT OF INFORMATICS

# **DATABÁZOVÝ SYSTÉM PRO OČNÍ OPTIKY**

DATABASE SYSTEM FOR OPTOMETRICIAN SHOPS

**BAKALÁŘSKÁ PRÁCE**  
BACHELOR'S THESIS

**AUTOR PRÁCE**  
AUTHOR

**LUKÁŠ DLOUHÝ**

**VEDOUCÍ PRÁCE**  
SUPERVISOR

Ing. JIŘÍ KŘÍŽ, Ph.D.

BRNO 2007

## Zadání bakalářské práce

## Zadání bakalářské práce

# LICENČNÍ SMLOUVA

## POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO

uzavřená mezi smluvními stranami:

### 1. Pan/paní

Jméno a příjmení: Lukáš Dlouhý

Bytem: Tábořská 428, Lomnice nad Popelkou

Narozen/a (datum a místo): 11.5.1983, Turnov

(dále jen „autor“)

a

### 2. Vysoké učení technické v Brně

Fakulta podnikatelská

se sídlem Kolejní 2906/4, 612 00, Brno

jejímž jménem jedná na základě písemného pověření děkanem fakulty:

Koch Miloš Doc. Ing. CSc.

(dále jen „nabyvatel“)

## Čl. 1

### Specifikace školního díla

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):

- ☐ disertační práce
- ☐ diplomová práce
- ☒ bakalářská práce
- ☐ jiná práce, jejíž druh je specifikován jako .....

(dále jen VŠKP nebo dílo)

Název VŠKP: Databázový systém pro oční optiky .....

Vedoucí/ školitel VŠKP: Ing. Jiří Kříž, Ph.D. ....

Ústav: Informatiky .....

Datum obhajoby VŠKP: Červen 2007 .....

VŠKP odevzdal autor nabyvateli v\*:

- ☒ tištěné formě – počet exemplářů .....1.....
- ☒ elektronické formě – počet exemplářů .....1.....

---

\* hodící se zaškrtněte

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.
3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.
4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

## **Článek 2**

### **Udělení licenčního oprávnění**

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizování výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti
  - ☒ ihned po uzavření této smlouvy
  - ☐ 1 rok po uzavření této smlouvy
  - ☐ 3 roky po uzavření této smlouvy
  - ☐ 5 let po uzavření této smlouvy
  - ☐ 10 let po uzavření této smlouvy(z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/ 1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

## **Článek 3**

### **Závěrečná ustanovení**

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísní a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne: .....

.....  
Nabyvatel

.....  
Autor

## **Abstrakt**

Práce se zabývá vytvořením vhodného databázového systému pro konkrétní společnost, která prodává a vyrábí brýle. Soustředí se na popis vývoje v jazyce SQL, HTML, PHP a také na strukturování databáze. Cílem práce je vytvořit databázový systém, který bude splňovat požadavky společnosti.

## **Klíčová slova**

Databáze, db, sql, html, php, MySql, systém, Internet, web, java-skript, xhtml, Apache, server, open source, SRŘDB, CSS, DML, DDL, DCL, hypertext, klient, SHA1, SSL, https, cookies, erd, formulář, optika, optometrista.

## **Abstract**

This thesis deals with creating a suitable database system for an specific existing company that manufactures and sells glasses. The thesis focuses on describing the development of the database system in SQL, HTML, PHP and also on the structuring of the database. It aims to create a database system, which will comply with the company's requirements.

## **Key words**

Database, db, sql, html, php, MySql, system, Internet, web, java-script, xhtml, Apache, server, open source, RDBMS, CSS, DML, DDL, DCL, hypertext, client, SHA1, SSL, https, cookies, erd, form, optic, optometrician.

## **Bibliografická citace mé práce**

DLOUHÝ, L. *Databázový systém pro oční optiky*. Brno: Vysoké učení technické v Brně, Fakulta podnikatelská, 2007. 50 s. Vedoucí bakalářské práce Ing. Jiří Kříž, Ph.D.

## **Čestné prohlášení**

Prohlašuji, že předložená diplomová práce je původní a zpracoval jsem ji samostatně. Prohlašuji, že citace použitých pramenů je úplná, že jsem v práci neporušil autorská práva (ve smyslu zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským, ve znění pozdějších předpisů).

V Brně, dne 18. května 2007

.....

Lukáš Dlouhý

## **Poděkování**

Děkuji vedoucímu bakalářské práce, panu Ing. Jiřímu Křížovi, Ph.D. za poskytnutí podnětných rad a připomínek. Dále pak vedení a zaměstnancům společnosti Sasfin a.s. za poskytnuté podklady a konzultace.



# Obsah

Abstrakt .....	6
Klíčová slova .....	6
Abstract.....	6
Key words.....	6
Bibliografická citace mé práce.....	7
Čestné prohlášení .....	7
Poděkování.....	8
Obsah.....	9
1. Úvod .....	11
2. Analýza problému a současné situace .....	12
2.1. Představení společnosti .....	12
2.2. Historie firmy .....	13
2.3. Stanovy společnosti – výťah .....	13
2.4. Analýza současného stavu.....	17
2.5. Shrnutí zjištěných nedostatků.....	18
3. Teoretická východiska práce.....	19
3.1. Poznatky .....	19
3.2. Nové trendy .....	20
3.3. MySQL.....	21
3.4. Jazyky .....	22
3.4.1. HTML .....	22
3.4.2. CSS.....	23
3.4.3. SQL .....	25
3.4.4. PHP .....	25
3.5. Architektura webové databázové aplikace.....	27
3.6. Bezpečnost .....	28
3.6.1. Možnosti šifrování hesel .....	28
3.6.2. SSL.....	28
3.6.3. Autentizace .....	29
4. Vlastní návrh řešení .....	30

4.1. Přihlašování.....	30
4.2. E-R diagram .....	31
4.3. Popis tabulek .....	31
4.4. Práce v systému .....	34
4.5. Struktura webového adresáře .....	36
4.6. Způsob generování stránek a obsahu .....	37
4.7. Formuláře .....	40
5. Přínosy navrhovaného řešení .....	42
5.1. Rychlost navrženého systému .....	42
5.2. Urychlení práce .....	42
5.3. Rozšiřitelnost systému .....	42
5.4. Přenositelnost .....	43
5.5. Nákladová stránka řešení .....	43
5.6. Kvantifikované přínosy .....	44
5.7. Nekvantifikované přínosy .....	44
6. Závěr .....	45
Seznam použitých zdrojů.....	47
Seznam internetových zdrojů.....	47
Seznam tištěné literatury.....	47
Seznam obrázků .....	48
Seznam příloh .....	48

# 1. Úvod

Společnosti, které mají více poboček a zabývají se prodejem a výrobou brýlí, potřebují ke své činnosti nějaké úložiště dat, aby měly po ruce aktuální informace o zákaznících, kteří přijdou z jedné pobočky do jiné. V současné době společnost nedisponuje programem, který by synchronizaci a získávání dat v reálném čase umožňoval. Nutnost takového úložiště způsobuje fakt, že přístup k informacím jedné pobočky o zákazníkovi druhé pobočky, bývá znemožněn absencí pracovníka, který právě někoho obsluhuje nebo v případě, že je jedna z provozoven ještě zavřená. Čekání společnost značně degraduje v očích zákazníka, snižuje zisky a poškozuje pohodu na pracovišti.

Informace, které se do databáze budou ukládat, jsou poměrně rozmanitého charakteru. Jedná se o zakázky, které již byly v minulosti se zákazníkem uskutečněny, kontroly zraku a údaje o naměřených hodnotách ať už se jedná o nová či starší měření.

Cílem práce je vyřešení problému synchronizace dat a zajištění přístupu k informacím v reálném čase prostřednictvím centralizované databáze záznamů o klientech. Po zavedení takového systému, by u společnosti mělo dojít k významnému snížení nákladů, způsobených telefonováním. Dále pak ke zvýšení renomé firmy a případné zvýšení zisků a počtu zákazníků z důvodu větší spokojenosti s poskytovanými službami. Zároveň s vyřešením těchto problémů se hodlám zabývat otázkou, jak nejlépe údaje interpretovat a dosáhnout snadné ovladatelnosti programu.

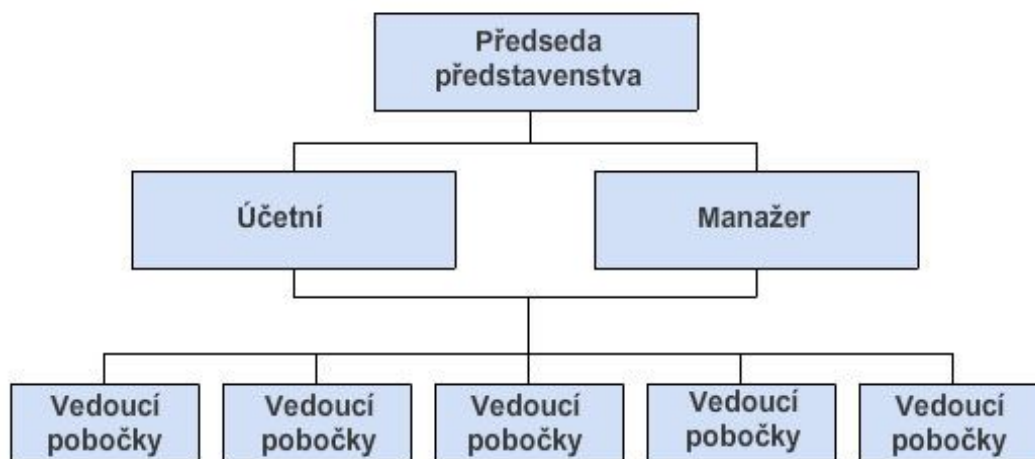
## 2. Analýza problému a současné situace

### 2.1 Představení společnosti

Společnost SASFIN a.s. se zabývá specializovaným maloobchodem. Jedná se především o výrobu a prodej brýlí a doplňkového sortimentu, včetně kontaktních čoček. V prodejní nabídce je široké spektrum brýlových obrub, od méně známých značek až po ty nejluxusnější. Společnost disponuje pěti pobočkami označenými logem „Apia“.

Všechny jsou velmi dobře strategicky umístěné. Dvě z nich leží na jednom z nejfrekventovanějších míst v Brněnském centru, na pěší zóně vedoucí z ulice Česká k hlavnímu nádraží. Další pobočku nalezneme v obchodním centru „Velký Špalíček“. V pořadí čtvrtou můžeme navštívit, také téměř v centru Brna, v obchodní galerii Vaňkovka. Poslední oční optika Apia je umístěna v prostorách nákupního centra Olympia, na jižním okraji města. Ačkoli je tato provozovna vzdálena, jedná se o jednu z nejvíce prosperujících.

V každé z poboček je zaměstnáno přibližně 5 zaměstnanců. V celkovém součtu všech zaměstnanců se jedná o malou až středně velkou organizaci.



Obr. 1 Hierarchie ve společnosti

## **2.2 Historie firmy**

V roce 1991 byla založena společnost Apia, s.r.o., která provozovala maloobchod a velkoobchod s brýlemi. Prodej brýlí a dalšího sortimentu konečným spotřebitelům se uskutečňoval prostřednictvím stávajících poboček na Náměstí Svobody a na Masarykově ulici v Brně. Čtvrtého června 1998 změnila právní formu na akciovou společnost a byla zapsána do obchodního rejstříku, pod názvem subjektu SASFIN a.s., kterou je doposud. Nyní se však zabývá pouze maloobchodním prodejem a velkoobchod přešel pod novou společnost APIA CZ, s.r.o.

V následujících letech firma expandovala a v prosinci roku 1999 otevřela třetí pobočku v nákupním centru Olympia. 31.8.2001 vznikla další oční optika v obchodním centru Velký Špalíček a jako poslední byla 23.3.2005 otevřena provozovna v nákupní galerii Vaňkovka. Ve výhledu do budoucna se společnost chystá expandovat i mimo území města Brna.

## **2.3 Stanovy společnosti – výtah**

### **Obchodní firma a sídlo společnosti:**

Obchodní firma: SASFIN a.s.

Sídlo společnosti: Brno, Běhounská 7, okres: Brno-město, PSČ 602 00

IČ: 25530895

### **Předmět podnikání (činnosti) společnosti**

- koupě zboží za účelem jeho dalšího prodeje a prodej
- velkoobchod
- specializovaný velkoobchod a maloobchod se smíšeným zbožím
- zprostředkovatelská činnost
- zprostředkování obchodu a služeb
- oční optika

### **Vznik a trvání společnosti**

Doba trvání společnosti není časově omezena.

## **Výše základního kapitálu, akcie**

Výše základního kapitálu společnosti činí 1 000 000,- Kč. Celý základní kapitál je rozdělen na celkem 100 ks listinných akcií znějících na majitele, každá z nich o jmenovité hodnotě 10 000,- Kč. Veškeré akcie společnosti a jejich emisní kurs, splatili zakladatelé jednorázově, v okamžiku založení společnosti, peněžitým vkladem. Každá akcie na majitele je neomezeně převoditelná. Vlastní převod akcie se uskutečňuje předáním předmětné akcie, případně akcií. Každé akci náleží jeden hlas na každých deset tisíc její jmenovité hodnoty.

## **Orgány společnosti**

- a) valná hromada
- b) představenstvo
- c) dozorčí rada

## **Valná hromada**

Valná hromada se musí konat minimálně jednou za kalendářní rok, nejpozději do šesti měsíců od posledního dne účetního období. Svolává ji představenstvo nebo jeho člen dle zákona. Představenstvo má povinnost uveřejnit konání valné hromady v celostátním deníku „Mladá fronta DNES“ nebo jiném, pokud již deník nevychází a to minimálně 30 dní před konáním valné hromady. Pokud chce akcionář nebo akcionáři vznést žádost k projednání, musí mít akcie, jejichž souhrnná hodnota přesahuje 5% základního kapitálu. Valná hromada je schopna se usnášet pouze v případě, že přítomní akcionáři disponují akciemi, jejichž jmenovitá hodnota je alespoň 50% základního kapitálu. Jestliže schopna usnášení není, svolá představenstvo novým oznámením další valnou hromadu a to nejpozději do 15 dní ode dne, kdy byla svolána původní valná hromada. Ta se musí konat do šesti týdnů ode dne předchozí valné hromady. Musí mít nezměněný pořad jednání a je schopna usnášení bez ohledu na počet účastníků se akcionářů.

## **Představenstvo**

Statutárním orgánem společnosti je představenstvo. Řídí její činnost, rozhoduje jejím jménem a vůbec rozhoduje o všech záležitostech společnosti, pokud nejsou zákonem nebo stanovami vyhrazeny do působnosti valné hromady nebo dozorčí rady. Zabezpečuje také obchodní vedení, včetně řádného vedení účetnictví a předkládá valné

hromadě ke schválení řádnou, mimořádnou, konsolidovanou, případně i mezitímní závěrku a návrh na rozdělení zisku nebo úhradu ztráty.

Představenstvo má tři členy, je voleno s výjimkou prvního představenstva, které je jmenováno zakladatelem. Funkční období představenstva je pět let a je možné znovuzvolení jeho členů. Členem může být fyzická osoba starší 18 let, která splňuje podmínky pro výkon funkce.

Zasedání představenstva svolává jeho předseda podle potřeby, nejméně jednou za čtvrt roku a nejpozději do 15ti dnů poté, co o svolání požádal kterýkoli člen nebo dozorčí rada. Pokud předseda tuto svoji povinnost nesplní, může představenstvo svolat, kterýkoli z jeho členů. Schopnost usnášet se nastává pouze tehdy, je-li přítomna nadpoloviční většina jeho členů. Rozhodují tedy alespoň dva hlasy. Pro rozhodování o zvolení nebo odvolání předsedy představenstva jsou nutné i zbývající dva hlasy jeho členů.

### **Dozorčí rada**

Dozorčí rada vykonává dohled nad výkonem působnosti představenstva a uskutečňováním podnikatelské činnosti společnosti. Její osazenstvo je oprávněno nahlížet do všech dokladů a záznamů týkajících se činnosti společnosti a kontroluje, zda jsou účetní zápisy vedeny řádně a v souladu se skutečností a zda se podnikatelská činnost společnosti uskutečňuje v souladu s právními předpisy, stanovami a pokyny valné hromady. Členové dozorčí rady se účastní valné hromady, kterou mohou i svolat a jsou povinni seznámit valnou hromadu s výsledky své kontrolní činnosti.

Stejně jako v případě představenstva, má dozorčí rada také 3 členy, kteří jsou voleni na dobu funkčního období pěti let a mohou být opětovně zvoleni. Tyto členy volí v tomto případě valná hromada, protože společnost nemá více než 50 zaměstnanců. Členem dozorčí rady nesmí být člen představenstva, prokurista společnosti nebo osoba oprávněná podle zápisu v obchodním rejstříku jednat jménem společnosti. Zasedání dozorčí rady svolává její předseda podle potřeby, nejméně však jednou za čtvrt roku a nejpozději do 15ti dnů poté, co o jeho svolání požádal kterýkoli člen dozorčí rady nebo představenstvo. V případě, že předseda dozorčí rady svoji povinnost nesplní, může dozorčí radu svolat kterýkoli její člen. Každý člen má jeden hlas a při zasedání je rozhodováno většinou hlasů.

## **Prokurista**

Prokurista zastupuje společnost v rozsahu stanoveném obchodním zákoníkem. Je jmenován do funkce představenstvem společnosti. K jeho jmenování resp. odvolání jsou potřebné dva platné hlasy členů představenstva. Představenstvu také odpovídá za svou činnost.

## **Rezervní fond společnosti**

Společnost vytváří rezervní fond z čistého zisku v řádné účetní závěrce za rok, v němž poprvé čistý zisk vytvoří, a to ve výši 20% z čistého zisku, avšak ne více než 10% z hodnoty základního kapitálu. Tento fond se ročně doplňuje o částku 5% z čistého zisku, a to až do zákonem požadované 20% výše základního kapitálu.

Takto vytvořený rezervní fond do výše 20% základního kapitálu lze použít pouze k úhradě ztráty.

O použití rezervního fondu rozhoduje představenstvo společnosti. Použití prostředků z rezervního fondu je přípustné teprve poté, co byly vyčerpány všechny ostatní prostředky k úhradě ztrát. O každém čerpání z rezervního fondu je představenstvo povinno podat zprávu dozorčí radě a valné hromadě.

Pro případ, že společnost rezervní fond použije k úhradě svých ztrát, musí jej do budoucna doplnit, a to tak, že ze svého čistého zisku vykázaného v řádné účetní závěrce za rok, v němž vytvoří zisk, převede do rezervního fondu 5% z tohoto čistého zisku, a to opět až do zákonem požadované 20% výše základního kapitálu.

## **Způsob rozdělení zisku**

Vyazuje-li roční účetní závěrka, po odečtení daní, čistý zisk, usnese se valná hromada, na návrh představenstva, na výši dotace jednotlivým fondům a roční části čistého zisku, která má být rozdělena mezi akcionáře formou dividend, a to vyjádřením jejich výše v procentech ze jmenovité hodnoty akcie a dále na výši tantiém.

Výše tantiém musí vždy dosáhnout, nejméně, výše dividend. Dividendy se poukazují akcionářům na jejich účet, udaný společností.

## **Možnosti úhrady ztrát společnosti:**

- a) z doposud případně nerozděleného zisku;
- b) čerpáním z rezervního fondu;
- c) snížením základního kapitálu;



d) *vydáním dluhopisů za podmínek uvedených v § 160 obchodního zákoníku.*

O způsobu úhrady ztrát společnosti rozhoduje, na návrh představenstva, valná hromada. Pořadí možného způsobu uhrazení ztrát společnosti je zcela volné.

## **2.4 Analýza současného stavu**

V současné době je společnost spíše rozděleným řetězcem poboček, bez možnosti rychlé výměny informací. Provozovny mají sice své stálé zákazníky, avšak často se stává, že klient z jedné pobočky navštíví pobočku na jiném místě. V tom případě zaměstnanci potřebují získat informace o provedených vyšetřeních, klientových zakázkách, přiznané slevy a atd. K uspokojení této potřeby se využívá telefonování prostřednictvím programu Skype nebo pevné linky. Tyto metody mají však svá úskalí.

Jedním z nich je různá otevírací doba jednotlivých poboček. V tomto případě jsou informace nedostupné a zaměstnanci se musí s klientem domluvit na pozdější návštěvě, případně zákazníka znovu vyšetřit, pokud se nalézá právě v pobočce, která to umožňuje. Vyšetření je však zatím možné provést pouze v určitém čase a není možné ho provést na všech pobočkách. Vyšetření zraku je poměrně zdoluhavým procesem a stává se, že zákazník, případně zaměstnanec nemá v danou chvíli dostatek času na jeho provedení.

Druhý problém nastává v případě, že zaměstnanci právě někoho obsluhují a nemohou se dostavit k telefonu. V tu chvíli, musí zákazník buď čekat nebo přijít, až se informace do pobočky dostanou, obvykle telefonicky nebo přes program Skype. O tom, že oční optika již informace má, je zákazník informován telefonicky, obvykle na mobilní telefon. Z tohoto procesu však vyplývají dodatečné náklady na komunikaci, které by mohly být společnou databází odstraněny.

**Příklad:**

Oční optika na Masarykově třídě má otevírací dobu od osmé hodiny. Zákazník, přichází v půl deváté ráno. Má zájem o kontaktní čočky, nepamatuje si typ, který používá a jeho vyšetřovací karta je na pobočce ve Vaňkovce. Vše potřebuje vyřídit v krátkém časovém intervalu neboť spěchá do práce.

Nastává problém. Obchodní galerie Vaňkovka otevírá v devět hodin. Proto jsou informace z optiky nedostupné.

Pracovnice v optice má několik možností:

1. hrát si na detektiva a cílenými otázkami zjistit zákazníkem používaný typ čoček, obvykle nereálné
2. omluvit se a poslat zákazníka na Vaňkovku po deváté hodině
3. v případě, že je zákazník ochoten čekat a vyzvednout si čočky třeba o polední pauze, může s ním být uzavřena zakázka, kdy zákazník skládá zálohu a pracovnice zjistí informace v otevírací době

Tento příklad může popisovat i situaci během celého dne, kdy na potřebné pobočce není nikoho, kdo by zvedl telefon, protože je zaneprázdněn.

Z uvedeného příkladu také vyplývá, jak je celá situace pro zaměstnance i pro zákazníka stresující.

**2.5 Shrnutí zjištěných nedostatků**

- absence aktuálních informací o zákazníkovi
- zvýšené náklady na komunikaci
- nadbytečné zaměstnávání pracovníků
- zvyšování stresové zátěže
- zdržování klientů
- speciální nároky na vybavení optiky
- pokles renomé v očích zákazníků

### 3. Teoretická východiska práce

Proto, aby mohly pobočky efektivněji spolupracovat je třeba vytvořit systém, ve kterém bude možné sdílet informace v reálném čase a bez nutnosti přímé komunikace pracovníků. Údaje, které chtějí pobočky sdílet jsou velice rozmanitého charakteru a nejvhodnějším prostředkem k výměně informací se zdá být centralizovaná databáze, přístupná odkudkoli a v jakémkoli čase. Všechny pobočky jsou již vybaveny dostatečně rychlým připojením k síti internet a z tohoto důvodu vidím, jako levné a vhodné médium, právě tuto síť.

Jaká data je třeba vlastně sdílet? Jak již bylo naznačeno v předchozí části práce, jedná se o data ze zakázek a vyšetření zákazníka. Bude se tedy jednat především o informace o trvalém bydlišti, přiznaných slevách, kontaktních číslkách, uskutečněných zakázkách na brýlové obruby a zábrusy čoček. Ukládané informace jsou tedy především textové a číselné údaje, případně ještě údaje časové.

#### 3.1. Poznatky

Celosvětová počítačová síť internet je v hojné míře využívána především z toho důvodu, že je výborným zdrojem informací, perfektním komunikačním prostředím, prostorem pro kreativní společnost a její využití je často zdarma. Díky řadě hackerů ochotných vyvíjet nové a vylepšovat stávající prostředky pro její rozmanitější využití, bez nároku na úplatu, stala se tato síť nezávislou na platformě uživatelů z celého světa. A právě přenositelnost softwaru je pro databázový systém vyvíjený v této práci nepostradatelnou vlastností.

Výstupy tohoto systému je nutné poskytovat zaměstnancům, pracujícím s různorodým hardware i s rozdílným programovým vybavením počítače. Splnitelnost tohoto požadavku je vázána na kód, ve kterém budou požadované informace předávány. Vzhledem k tomu, že na pobočkách optik je obvykle používán operační systém společnosti Microsoft na platformě Windows, pro který existuje nespočetné množství aplikací umožňujících interpretaci jazyka HTML, sympatizují se skriptovacím jazykem, který výsledky právě v tomto jazyce nabízí. Značkový jazyk HTML je navíc množstvím interpretů, schopných jeho zpracování možné využít i na dalších operačních systémech.

Hojně používaným skriptovacím jazykem, interpretujícím výsledky v HTML je jazyk PHP. Jeho schopnosti jsou rozšířeny o možnost vkládání skriptů přímo do kódu HTML. Jazyk je navíc vhodný i na vyžadované dotazy na databázi a dělá databázový server s konkurencí o poznání rychlejší pokud je nainstalován jako modul serveru Apache.

Právě tento server je podle serveru [www.netcraft.com](http://www.netcraft.com) číslem 1 při tvorbě webových databází. V kombinaci s jazykem PHP a databází MySQL tvoří konkurenceschopnou variantu k serverům společnosti Microsoft a jeho jazyka ASP.

Nespornou výhodou těchto prostředků k vyvíjenému systému je, že jsou k užívání zdarma. Jejich otevřený kód, na kterém pracují milióny lidí z open source komunity ho činí bezpečnějším než jiné varianty, na kterých pracuje za zavřenými dveřmi pár zasvěcených programátorů a je otestován několika beta-testery.

Bezpečnost těchto technologií spočívá i v častějších aktualizacích. Jakmile někdo z open source komunity objeví chybu, může vytvořit záplatu a tu ihned prostřednictvím internetu distribuovat uživatelům. O opravách a vylepšení programů komunita hojně diskutuje na diskusních fórech k tomu, kterému programu určenému a na dalších diskusních fórech fanoušků programu. Oproti komerčně vyráběnému software, má uživatel možnost nejen podílet se na tvorbě a ladění programu, ale i na jeho dalším rozvoji a přizpůsobit si ho na míru. Snadnosti úprav napomáhají nejen diskusní fóra, ale i komentáře v kódu programu.

### **3.2. Nové trendy**

Nové trendy v tomto vývoji usměrňuje open source komunita a programátorské společnosti tím, co vytváří a jak standardizační organizace přijímají nové standardy. Dle posledních novinek ve světě PHP, se zdá, že programování v tomto jazyce bude mít novou tvář. Do nedávné doby bylo programování v tomto jazyce především imperativního charakteru, nicméně s nově standardizovanou verzí jazyka ve verzi pět, přišlo plně podporované objektově orientované programování, nahrazující jeho dřívější podobu.

### 3.3. MySQL

Jedná se o nejrozšířenější open source databázi na světě. Tento databázový systém umožňuje technologiím PHP a Apache spolupracovat při zpřístupnění a zobrazení dat ve formátu čitelném webovými prohlížeči. Je to databáze zpracovávající dotazy ve strukturovaném dotazovacím jazyce (Structured Query Language - SQL) navrženém pro zpracování velkého množství složitých dotazů.

Jedná se o „systém správy relačních databází“ (*RDBMS, relational database management systém*) v češtině známý jako „*RSŘBD, relační systém řízení báze dat*“. MySQL tedy umožňuje ukládat samostatné informace do tabulek obsahujících související data a následně s nimi pracovat. Díky tomu nabízí maximální efektivitu a rychlost, oproti nerelačním databázovým systémům, v nichž jsou informace ukládány v jedné obrovské oblasti dat a operace nad nimi jsou těžkopádnější a složitější. Tabulky v této databázi jsou jednoznačně identifikovány názvem a obsahují sloupce v nichž jsou uloženy jednotlivé části informace.

#### **Základní a zároveň nejoblíbenější funkce:<sup>1</sup>**

- a) „Možnost využití většího počtu procesorů prostřednictvím vláken jádra,
- b) možnost spouštění nezávisle na hostitelské platformě,
- c) četné typy sloupců, pokrývající prakticky veškeré datové typy,
- d) skupiny funkcí pro matematické výpočty a třídění,
- e) příkazy, které umožňují snadný přístup správců k informacím o databázích,
- f) názvy funkcí, jež nekolidují s názvy tabulek ani sloupců,
- g) systém ověřování uživatelů na základě hesla,
- h) možnost definice 32 indexů pro každou tabulku; tato funkce byla s úspěchem implementována na úrovni 60 000 tabulek a 5 000 000 000 řádků,
- i) mezinárodní hlášení chyb použitelné v mnoha různých státech.“

---

<sup>1</sup> Zdroj: PHP5, MySQL, Apache – Vytváříme webové aplikace, str. 27

## 3.4. Jazyky

### 3.4.1. HTML

Zkratkou HTML je označován značkovací jazyk pro hypertext. Jedná se o jeden z několika jazyků, jehož prostřednictvím je možné publikovat webové stránky na Internetu. HTML je zkratkou z anglického názvu „HyperText Markup Language“. Autorem tohoto značkovacího jazyka, který vychází z univerzálního značkovacího jazyka SGML, je Tim Berners-Lee z výzkumného centra fyziky, CERN. Tento jazyk byl vyvinut za účelem snadnější výměny informací mezi vědci. Současně s ním byl vyvinut i protokol HTTP, prostřednictvím něhož měla být data přenášena v síti. O rok později CERN spustil svůj první web pro který bylo třeba vytvořit nějaký prohlížeč. Jednalo se o první prohlížeč s grafickým rozhraním jehož název je Mosaic. Byl vytvořen v roce 1993 a jeho úspěch odstartoval velký rozvoj webu a nutnost standardizace jazyka.

Jedná se vždy pouze o textový formát a vyskytují-li se na stránce nějaká binární data, jako například obrázky nebo animace, je na ně odkázáno, takže nejsou umístěna do souboru se základním popisem stránky. Všechny příkazy a jejich případné parametry jsou uzavírány do špičatých závorek. Vždy hned za první špičatou závorkou je umístěno jméno příkazu, a dále následují jeho parametry. Tyto příkazy jsou buď párové nebo nepárové. Párové slouží k formátování elementu. Jeho první díl je před formátovaným elementem a druhý těsně za ním, čím je vymezena oblast, na kterou se formátování aplikuje. Takovýmto způsobem funguje například tag FONT. Užívá se následujícím způsobem:

```
<FONT ALIGN="center">formátovaný text</FONT>
```

Nepárový příkaz se vztahuje na celý dokument nebo element, který už je sám osobě přesně vymezený. Příkladem nepárového příkazu může být tag <IMG>.

Např.: <IMG SRC="obrazek.jpg" WIDTH=100 HEIGHT=200>

#### Poslední verze - 4.01<sup>2</sup>

*„Byla vydána 24. Prosince 1999. Tato verze opravuje některé chyby verze předchozí a přidává některé nové tagy. Je to poslední verze HTML, které se již dále nevyvíjí, protože má být nahrazeno novějším XHTML, jehož základem je právě tato*

---

<sup>2</sup> Zdroj: <http://cs.wikipedia.org/wiki/Html>

*poslední verze HTML. Hlavním rozdílem mezi HTML a XHTML je uzavírání všech používaných tagů.“*

### **3.4.2. CSS**

Interpretované informace v prohlížeči, mohou být formátovány již samotnými tagy umístěnými v kódu jazyka HTML. Nicméně formátování mnoha stránek a jejich případná hromadná změna by byla časově velice náročná. Nejen z tohoto důvodu byl vyvinut systém formátování, který umožňuje změny ve formátu textu jediným skriptem a označuje se termínem „kaskádové styly“ a zkratkou CSS, „Cascading Style Sheets“. Dalším důvodem, proč bylo třeba styly zavést, je snaha oddělit v dokumentu obsažené informace od formátovacích informací, pro snazší vyhledávání a zpracování dokumentů. Styly uvedla a poprvé také implementovala společnost Microsoft v roce 1996. První prohlížeč, který je tehdy podporoval byl Internet Explorer a to ve verzi 3. V současné době existují specifikace pro verze CSS1, CSS2.1 a nejnovější stále vyvíjenou verzí je CSS3<sup>3</sup>. Základní koncept CSS je velice jednoduchý, lze totiž modifikovat výchozí vlastnosti stylů, které jsou již v HTML obsaženy, např. nadpisy (H1, H2, ...) a je také možné nové styly vytvářet a přiřazovat jim vlastnosti.

Kaskádové styly lze zapisovat přímo do HTML kódu stránky, avšak nejvhodnějším způsobem pro tento projekt je, aby byly uloženy v externím souboru a z něj načítány. Obsah stránek bude generován prostřednictvím PHP kódu a stačí tedy využít vloženého odkazu na styl v šablonách vytvořených pro jednotlivé stránky výstupů a PHP kódem vložit HTML obsah s třídami a identifikátory.

Z následujícího příkladu jsou patrné některé možnosti formátování webové stránky prostřednictvím kaskádových stylů. Rozdíl mezi identifikátorem ID a třídou CLASS spočívá nejen v zápisu, ale také v použití. Co se týče identifikátoru, ten by měl být jedinečný, tedy použit pouze jednou. Zatímco třída může být definována jako obecná nebo ji lze omezit pro formátování mezi určitými tagy.

---

<sup>3</sup> Zdroj: Organizace W3C, [www.w3.org](http://www.w3.org)

### Příklad odkazu na externí soubor CSS v HTML a formátování<sup>4</sup>:

```
<HTML>
<HEAD>
<title>Pokusná stránka</title>
<style type="text/css">@import url( default.css ); </style>
</HEAD>
<BODY>
<P id=zvyrazneni>Text upravený identifikátorem.</P>
<H3 id=podtrzeno>Text upravený identifikátorem.</H3>
<P class=cervene>Text upravený třídou.</P>
<P class=modre>Text upravený obecnou třídou.</P>
<H2 class=modre>Nadpis upravený obecnou třídou.</H2>
</BODY>
</HTML>
```

### Obsah souboru default.css:

```
/* Seskupování definicí */
P, H2 { font-family: Verdana, sans-serif }
/* Použití obecného identifikátoru */
#zvyrazneni { background: #FF0000 }
/* Omezení identifikátoru na tag H3 */
H3#podtrzeno { text-decoration: underline }
/* Omezení třídy na tag P */
P.cervene { color: #FF0000 }
/* Obecná třída */
.modre { color: #0000FF }
/* Použití pseudo elementu */
P:first-letter { font-size: 150% }
```

### Výsledek v prohlížeči Internet Explorer 6:

**T**ext upravený identifikátorem.

Text upravený identifikátorem.

Ttext upravený třídou.

Ttext upravený obecnou třídou.

**Nadpis upravený obecnou třídou.**

Obr. 2 Formátování CSS

---

<sup>4</sup> Tento příklad by zpracován na základě informací z knihy: Vytváříme WWW stránky a spravujeme moderní website



### 3.4.3. SQL

Dotazovací jazyk SQL, pro práci s daty v relačních databázových systémech, je přejmenovaným původním dotazovacím jazykem SEQUEL, který vytvořila společnost IBM. V současné době je nejnovější verzí tohoto jazyka verze SQL3 (SQL-99) standardizovaná americkým institutem ANSI. „*Standardy podporuje prakticky každá relační databáze, ale obvykle nejsou implementovány vždy všechny požadavky normy. A naopak, každá z nich obsahuje prvky a konstrukce, které nejsou ve standardech obsaženy. Přenositelnost SQL dotazů mezi jednotlivými databázemi je proto omezená.*“<sup>5</sup>

SQL je založeno na množinových a relačních operacích. Typickým příkladem dotazu je příkaz pro výběr:

**„select seznam atributů from seznam relací where podmínka“**

Jedná se o příkaz pro manipulaci s daty, mezi které patří i další dotazy (INSERT, UPDATE, DELETE, atd.). Tyto příkazy se označují zkratkou **DML** – „*Data Manipulation Language*“.

Další skupina příkazů slouží k tvorbě struktury databáze. Jedná se o příkazy, CREATE, ALTER a DROP. Tato skupina nese zkrácené označení **DDL** – „*Data Definition Language*“. Příkladem použití může být tvorba tabulky:

**„CREATE TABLE `test` (`id` TINYINT NOT NULL AUTO\_INCREMENT, PRIMARY KEY (`id`));“**

Do poslední skupiny patří příkazy pro řízení přístupových práv a řízení transakcí. Jsou označovány jako **DCL** – „*Data Control Language*“, („jazyk pro ovládání dat“). Mezi tyto příkazy patří: GRAND, REVOKE, COMMIT, ROLLBACK a BEGIN.

Existuje ještě řada dalších příkazů, především pro správu databáze. S velkým množstvím z nich se můžeme setkat, například v open source projektu pro správu databáze, PHPMyAdmin.

### 3.4.4. PHP

Hypertextový preprocesor PHP je skriptovacím jazykem bohatě vybaveným pro tvorbu dynamických webových stránek i aplikací běžících na počítačích lokálně. Jeho

---

<sup>5</sup> Zdroj: Otevřená encyklopedie Wikipedia, <http://cs.wikipedia.org/wiki/SQL>

kód je nezávislý na platformě a je tedy přenositelný. O vývoj jazyka se opět stará open source komunita, takže je poskytován zdarma a lze ho přizpůsobit svým potřebám, případně si vytvořit záplatu na nově objevenou chybu. PHP lze nainstalovat jako modul serveru Apache, čím se podstatně zvýší rychlost zpracování požadavků. Kód skriptů se zpracovává na straně serveru a ten uživateli vrací kód v jazyce html. Je tedy zaručena neviditelnost a tím pádem i bezpečnost kódu skriptu. Samotný kód skriptu v php lze začlenit do kódu jazyka html, čehož se velmi často využívá, například pro generování obsahu šablon. Součástí jazyka jsou i rozsáhlé knihovny sloužící ke zpracování textu, grafiky, práce se soubory, podpoře celé řady internetových protokolů a pro přístup k velkému množství databázových systémů. Syntaxe jazyka v sobě kombinuje několik starších jazyků (C, Perl, Javu a Pascal), což usnadňuje programátorům, kteří s ním začínají, jeho vstřebání.

#### **Příklad kódu:**

```
<?php
    echo „Text generovaný skriptem v jazyce PHP“;
?>
```

#### **Historie:<sup>6</sup>**

*„Od roku 1994 je PHP jedním z nejpoužívanějších způsobů tvorby dynamicky generovaných WWW stránek. Jeho tvůrce (Rasmus Lerdorf) jej vytvořil pro svou osobní potřebu přepsáním z Perlu do jazyka C. Sada skriptů byla vydána ještě v téže roce pod názvem Personal Home Page Tools, zkráceně PHP.*

*V polovině roku se systém PHP spojil s programem Form Interpreter stejného autora. Tak vzniklo PHP/FI 2.0. Zeev Suraski a Andi Gutmans v roce 1997 přepsali parser a zformovali tak základ PHP3. Současně byl název změněn na dnešní podobu PHP hypertext procesor. PHP3 vyšlo v roce 1998, bylo rychlejší, obsahovalo více funkcí. Také běželo i pod operačním systémem Windows.*

*V roce 2000 vychází PHP verze 4, o čtyři roky později pak verze 5 s vylepšeným objektovým přístupem, podobným jazyku Java. Zatím poslední verzí je 5.2.1 z roku 2007.“*

#### **Typické vlastnosti jazyka PHP:**

*„Jazyk PHP je dynamicky typový, tzn. že datový typ proměnné se určí v okamžiku přiřazení hodnoty. Díky tomu má PHP dva typy porovnání, '==' stejný jako v C, a*

---

<sup>6</sup>Zdroj: <http://cs.wikipedia.org/wiki/PHP>

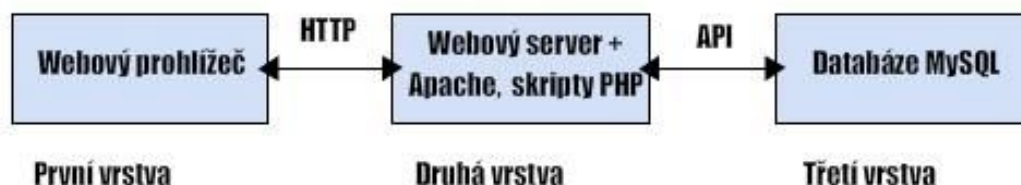
'===' který platí jen když jsou oba dva výrazy stejného typu. Pole jsou heterogenní, mohou tedy obsahovat jakékoli údaje, stejně tak jako jejich indexy. Řetězce lze uzavírat jak do uvozovek (obsah je parsován), tak do apostrofů (obsah není parsován). PHP do verze 4.2.0 automaticky přejímalo veškeré proměnné (považovalo je za globální) poslané jakoukoliv metodou (POST, GET, COOKIE, ale i SESSION) (pokud administrátor serveru tuto funkci nevypnul) a umožňovalo s nimi dále pracovat. Od verze 4.2.0 lze hodnotu získat ze superglobálních proměnných s garancí původu informace (máme zaručeno, že data byla odeslána požadovanou metodou). Administrátor serveru má stále možnost používání globálních proměnných povolit, ale z bezpečnostních důvodů se to silně nedoporučuje.“

### 3.5. Architektura webové databázové aplikace

Hlavní myšlenkou při zpracování této bakalářské práce je vývoj systému, který bude mít třívrstvé uspořádání. Základními částmi webové aplikace obvykle jsou:<sup>7</sup>

- a) „klient: uživatelův webový prohlížeč, java applet, aplikace java nebo dokonce na platformě závislý klientský program;
- b) logika aplikace: zakódovaná do algoritmů používaných ve skriptech CGI, speciálních modulech webového serveru nebo dokonce v serveru závislém na aplikaci;
- c) databázová konektivita: API databáze nebo obecné propojovací protokoly, jako např. ODBC nebo JDBC;
- d) databázový server: RDBMS, ODBMS atd.“

V tomto případě bude uspořádání vrstev následující:



Obr. 3 Třívrstvá architektura projektu

<sup>7</sup> Zdroj: Programujeme PHP profesionálně, str. 224

## 3.6. Bezpečnost

Databázový systém bude obsahovat řadu citlivých informací nejen o zákaznících, ale i o obchodní činnosti společnosti, které mohou být zneužity třetí stranou. Na údaje o zákaznících se navíc vztahuje zákon o ochraně osobních údajů. Z tohoto důvodu je nutné zvážit možnosti, jak aplikaci zabezpečit tak, aby nedocházelo k úniku informací.

### 3.6.1. Možnosti šifrování hesel

Programovací jazyk PHP obsahuje velké množství hašovacích funkcí pro šifrování hesel. Jejich funkce a implementace do databázového systému bude řešena ve vlastním návrhu řešení. Zde uvádím výběr funkcí z knihovny mhash, vhodných pro použití.

S přihlédnutím ke vzrůstajícímu výkonu výpočetní techniky je nutné zabývat se otázkou, jak dlouhý hash je nutný pro bezpečné zašifrování hesel umístěných v databázi. V současné době se doporučuje<sup>8</sup> využívat šifrovací algoritmy, jejichž výstupem je hash dlouhý alespoň 160 a více bitů. Tomuto standardu vyhovují například tyto funkce<sup>9</sup>:

- *SHA1, SHA160, SHA192, SHA224, SHA384, SHA512, HAVAL160, HAVAL192, HAVAL224, HAVAL256, RIPEMD256, RIPEMD320, TIGER160, SNEFRU256*

### 3.6.2. SSL

Vzhledem k tomu, že jsou PHP skripty zpracovávány na straně serveru, tak lze odposlouchávat informace zadávané do formulářů v nešifrované podobě a případný útočník si je může přečíst, bez využití dalších prostředků pro dešifrování. Tuto možnost je třeba vyloučit. K řešení tohoto problému se nám nabízí využití šifrované komunikace prostřednictvím SSL.

*„SSL je zkratka z anglického Secure Sockets Layer. Je to protokol/vrstva, vložená mezi vrstvu transportní (např. TCP/IP) a aplikační (např. HTTP). Poskytuje zabezpečení komunikace šifrováním a autentifikací komunikujících stran.*

**Princip:**

---

<sup>8</sup> Zdroj: <http://www.pcsvet.cz/art/article.php?id=4590>

<sup>9</sup> Zdroj: <http://mhash.sourceforge.net/>

*Ustavení SSL spojení funguje na principu asymetrické šifry, kdy každá z komunikujících stran má dvojici šifrovacích klíčů - veřejný a soukromý. Veřejný klíč je možné zveřejnit a pokud tímto klíčem kdokoliv zašifruje nějakou zprávu, je zajištěno, že ji bude moci rozšifrovat jen majitel použitého veřejného klíče svým soukromým klíčem.*<sup>10</sup>

### **3.6.3. Autentizace**

K ověřování autentičnosti uživatelů je možné využít několik metod. Jednou z často používaných a prověřených, je ověřování přihlašovacího jména a hesla. K často využívaným metodám pro tento systém, je ověřování prostřednictvím uživatelské relace a souboru cookie. Skrze tyto soubory můžeme zajistit, že přístup k jednotlivým stránkám webu a využívání některých funkcí, budou mít jen autorizovaní uživatelé. Pro ještě lepší zabezpečení je možné zkombinovat tuto metodu například s využitím souborů htaccess, jejichž samostatné využití není příliš vhodné pro tento systém.

#### **Ve zkratce:**

*„Cookies jsou malé textové informace, které se dají uložit na počítači klienta. Z bezpečnostních důvodů je ukládání všech souborů na klientovi zakázáno, cookies jsou jedinou omezenou výjimkou. Ukládají se někde do cache (nejčastěji do dočasných souborů), lze se na ně podívat přes možnosti prohlížeče. Lze je vypnout (zakázat ukládání) a smazat.*

*Ukládat a číst cookies umějí skripty klientské i serverové. Každé cookie má název a hodnotu plus informaci o vypršení a rozsah platnosti.*

*Pokud se nenastaví jinak, může skript cookies číst jen z té stránky, na které bylo uloženo. Při ukládání se může nastavit path a domain, potom půjde číst i ze stránek zadané cesty a domény. Jestliže se ale nastaví domain jiné, než jaká je aktuální doména (druhého řádu), nebude to fungovat.*

*V případě, že by se cookie snažil nastavovat jiný objekt ve stránce (např. obrázek), který se ale do stránky načítá z jiné domény, jde o tak zvanou cookie třetí strany (third part cookie). Na některých prohlížečích je ukládání takových cookies omezeno nebo zakázáno -- záleží to na nastavení prohlížeče a na tom, zda server má P3P soubor.*<sup>11</sup>

---

<sup>10</sup> Zdroj: <http://cs.wikipedia.org/wiki/SSL>

<sup>11</sup> Zdroj: [www.jakpsatweb.cz/enc/cookies.html](http://www.jakpsatweb.cz/enc/cookies.html)

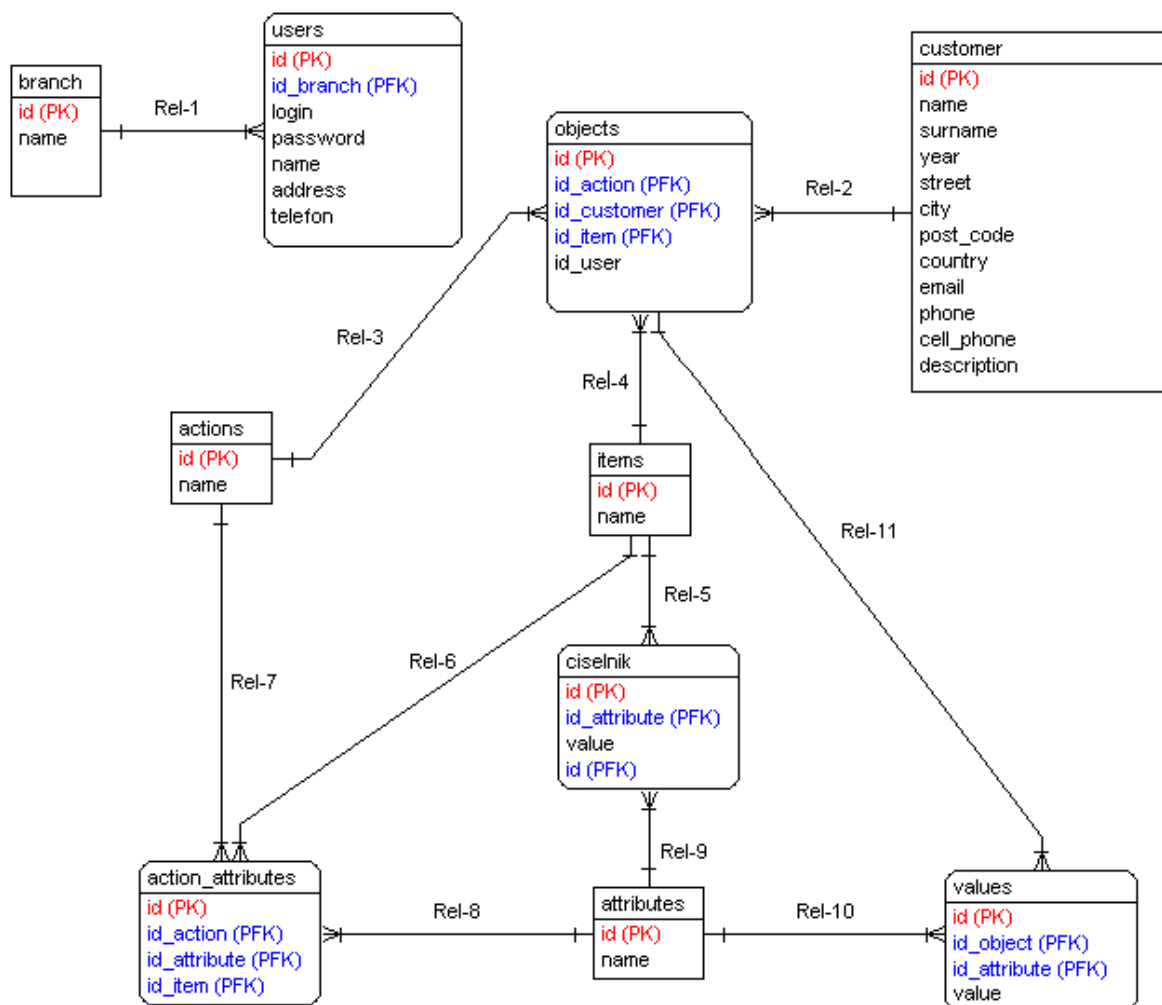
## **4. Vlastní návrh řešení**

Při vývoji webové aplikace musíme projít řadou kroků, než se dostaneme k ostrému nasazení produktu ve společnosti. V první řadě je nutné definovat daný problém a specifikovat jeho zadání. Jeho nástin byl již částečně proveden v předchozí části práce. Dále je třeba analyzovat systém jako takový a specifikovat seznam požadavků, které jsou na nás při vývoji systému kladeny. Poté můžeme začít s návrhem struktury a architektury aplikace. Jedná se především o entitně-relační diagram, diagram datových toků a návrh tabulek obsažených v databázi. Po ukončení této důležité fáze přijde na řadu tvorba kódu a sestavení aplikace, jeho doladění, testování a na závěr jeho implementace v prostředí internetu.

### **4.1. Přihlašování**

Každá pobočka má přiřazen jeden vlastní login a heslo. Tento systém byl přijat z důvodu přechodů brigádníků a stálých zaměstnanců mezi jednotlivými pobočkami. Díky tomuto rozdělení je možné vyhledávat informace o zakázkách z jedné nebo více poboček a tvořit k nim statistiky prodeje. Identifikační číslo pobočky je tedy přidělováno ihned při přihlášení a uživatel vystupuje jako pracovník identifikované pobočky. Údaj o tom, ze kterého pracoviště uživatel přistupuje k systému je zobrazen v hlavičce jednotlivých stránek, aby nedošlo k záměně.

## 4.2. E-R diagram



Obr. 4 Entitně-relační diagram

## 4.3. Popis tabulek

### Tabulka *users*

Tato tabulka slouží k uložení informací o přihlašování. Jedná se o uživatelská jména a hesla zaměstnanců. Tyto informace jsou z důvodu zvýšení bezpečnosti ukládány zašifrovaně prostřednictvím hašovací funkce SHA1 s délkou haše 160 bitů. Dále jsou v tabulce uloženy informace o pobočce (název, adresa a telefon), které slouží k rychlejší tvorbě reklamačních formulářů.

### **Tabulka *branch***

Obsahem této tabulky je název, pobočky a prostřednictvím atributu ID (auto-increment) je tento název spojován s přihlášeným uživatelem. Rozdělení tabulek users a branch bylo vytvořeno z důvodu možného sledování zaměstnanců v jednotlivých pobočkách.

### **Tabulka *customer***

Stěžejní informací identifikující zakázku, jsou informace o zákazníkovi. Jeho identifikační číslo provází každou vytvořenou zakázku. Stejně tak, jako obsah atributu description. Tento atribut skrývá poznámky k samotnému zákazníkovi, které jsou u kterékoli zakázky stejné a je možno je doplňovat. Jedná se o text obsahující například solventnost zákazníka, některé jeho vlastnosti, na které by si případný prodejce měl dát pozor. Mezi další atributy patří již tradičně, jméno a příjmení, dále pak informace o adrese zákazníka. Jedná se tedy o ulici, město, poštovní směrovací číslo a stát (zákazníky jsou i cizinci). Nalezneme zde i kontaktní informace, mezi ně patří telefonní číslo pevné linky, mobilní telefonní číslo a e-mailová adresa zákazníka.

Všechny tyto informace jsou zadávány do formuláře, při tvorbě nového zákazníka.

### **Tabulka *actions***

V tabulce jsou přiřazena jednotlivá id k hodnotám pro volbu akce. Akce jsou rozděleny na prodej, reklamaci a kontrolu. Tato tabulka slouží pro případné rozšíření systému o další akce do budoucna.

### **Tabulka *items***

Pro volbu sortimentu slouží právě tato tabulka. Obsahuje opět identifikační číslo, dále pak název (brýle, sluneční brýle, kontaktní čočky a brýlová skla). Tabulka obsahuje všechny druhy zboží, které bude systém vést v evidenci.

### **Tabulka *inventory***

Obsah tabulky tvoří klíč id, který je nastaven jako auto-increment, dva cizí klíče (id\_item, id\_attribute) a samotná hodnota položky s atributem value. Hodnotou atributu value jsou jednotlivé značky brýlí a kontaktních čoček. O tom, že se jedná o brýle nebo kontaktní čočky, případně jiné zboží, zjistíme informaci z cizího klíče id\_item, zatímco



cizí klíč `id_attribute` slouží k přiřazení vlastnosti z tabulky `attributes`. Hodnoty jsou využity při administraci seznamů a pro urychlení zadávání dat do formulářů tím, že se některá z nich vybere z vygenerovaného seznamu. Z dat, které obsahuje jsou tvořeny všechny číselníky v systému.

#### **Příklad dat v tabulce inventory**

<b>id</b>	<b>id_item</b>	<b>id_attribute</b>	<b>value</b>
1	4	37	Antifatique Stylis 1,67 CAL
Využito při zobrazení číselníku.	Jedná se o brýlová skla.	Jedná se o typ brýlových skel.	Značka skel.

#### **Tabulka objects**

Jedná se o spojovací a jednu ze stěžejních tabulek celé databáze. Sdružuje dohromady jednotlivé klíče z tabulek `users`, `customer`, `items` a `actions`. Tabulka neobsahuje konkrétní hodnoty údajů ze zakázek, ale eviduje všechny činnosti, předměty a zákazníky. Jedná se například o propojení: prodej, zadán od někoho, pro určitého zákazníka.

#### **Tabulka action\_attributes**

Další z vazebních tabulek je `action_attributes`. Slouží k přidělení atributů k různým akcím a předmětům. Data, uložená v této tabulce, jsou základem k formulářovým oknům při zvolené akci. Jestliže generujeme formulářové okno pro zakázku brýlí, pak zjistíme požadované formulářové vstupy právě v této tabulce.

#### **Tabulka attributes**

Jedná se o číselník všech vlastností, které jsou využity při tvorbě formulářů. Uspodňuje práci při zadávání nových dat do formulářů a ukládání do databázového systému.

#### **Tabulka values**

V této tabulce jsou uložena všechna data o zakázkách, formou přiřazení hodnoty k objektu a k určitému atributu.

### Příklad hodnot uložených v tabulce values

<i>Id</i>	<i>id_attribute</i>	<i>id_object</i>	<i>value</i>
1	10	1	B8008
Identifikátor	Číslo vlastnosti z tabulky attributes	Číslo objektu z tabulky objects	V tomto případě, údaj o typu brýlí

## 4.4. Práce v systému

### Přihlašování

Při vstupu na webové rozhraní databázové aplikace se jako první zobrazí přihlašovací formulář. V tomto formuláři systém žádá o zadání uživatelského jména a hesla. Jestliže zadáme správné údaje, systém nás přesměruje na úvodní stránku. V případě, že jsou přihlašovací informace nesprávné, pak systém zobrazí zprávu o špatně zadaných údajích a přesměruje stránku samu na sebe.

### Rozcestník

Když se dostaneme na úvodní stránku, můžeme v levém sloupci, začít vyhledávat již vloženého zákazníka. Obsahem hlavního okna je rozcestník. Zde jsou umístěna tlačítka pro volbu další akce. Stisknutím tlačítka se dostaneme na stránku zvolené akce. Jedná se o formulář sloužící k tvorbě nového zákazníka, stránku administrace a stránku pro tvorbu specifických dotazů.

### Tvorba nového zákazníka

Jestliže zvolíme funkci „nový zákazník“ dostaneme se na stránku s formulářem. Na této stránce můžeme zvolit i další akci, kterou zákazníkovi přiřadíme. Výběr akce umožňují seznamy v levé části. Po výběru akce formulář uložíme a systém nás přesměruje přímo na formulářové okno, patřící té, které akci. Díky tomuto řešení zkrátíme dobu, kterou bychom potřebovali při opětovném vyhledání zákazníka a přiřazení akce.

### Administrace

Administrační stránka slouží k přidávání nových typů brýlí, skel a kontaktních čoček do seznamů, které jsou využity pro rychlejší a přesnější zadávání údajů v zakázkách. Okno obsahuje výběrový seznam typů produktů a kolonku, do které se zapisuje nová značka obrub, typ kontaktních čoček nebo skel do brýlí. K odeslání údaje

a jeho uložení do databáze, slouží tlačítko „přidat“. Tyto přesné údaje lze využít například při tvorbě statistik prodeje jednotlivých značek a typů zboží na jednotlivých pobočkách nebo hromadně.

### Využití funkce „vyhledat zákazníka“

Do vyhledávacího formuláře stačí zadat pouze jedno písmenko ze jména hledaného zákazníka. Po stisknutí tlačítka hledej, zobrazí systém seznam zákazníků, které tento znak obsahují. Tento seznam je tvořen odkazy ve formátu příjmení, jméno, ulice a město. Tyto informace stačí k identifikaci. Jestliže chceme pouze upravit informace o zákazníkovi, pak se na jeho formulář dostaneme jediným kliknutím. Když chceme přidat zákazníkovi novou akci (prodej, apod.), poté stačí zvolit tzv. „radio button“ u jména. Po výběru zvolíme akci a druh zboží ze seznamu pod výpisem a požadavek odešleme. Systém nás přesměruje přímo na daný formulář a můžeme začít se zadáváním údajů.

### Javascript, použitý ve formulářích

Pro zadávání datumu, v jednotném formátu, jsem využil již vytvořené javascriptové funkce. Po kliknutí myši do pole pro datum je zobrazen kalendář, ve kterém je implicitně nastaveno aktuální datum. Po zvolení požadovaného data je políčko vyplněno.

V tomtéž java-skriptovém souboru jsem vytvořil další funkci, která slouží jako jednoduchá kalkulačka. Jedná se o pole týkající se plateb za zakázku. Tato část formulářového okna pro její tvorbu je na obrázku dole.

Platba			
Obruba	<input type="text"/>	Celkem	<input type="text"/>
Levá čočka	<input type="text"/>	Záloha	<input type="text"/>
Pravá čočka	<input type="text"/>	Recept	<input type="text"/>
Sleva brýle	<input type="text"/>	Kartou	<input type="checkbox"/>
		Sleva čočky	<input type="text"/>
Zůstatek	<input type="text"/>		

Obr. 5 Kalkulátor plateb

V této části formuláře se vyplňují pouze částky v kolonkách obruba, levá čočka, pravá čočka. Případně se ještě zadávají údaje do políček slev. Skript automaticky dopočítává celkovou částku a vkládá ji do políčka Celkem. Z této částky se po vložení dat do vstupu recept nebo záloha přepočítá částka a vloží se do položky zůstatek. Jedná se o sumu, která má být ještě zaplacená.

### Ostatní funkce systému

Kromě zobrazování přihlášeného uživatele a jeho příslušnosti v pobočce, najdeme v záhlaví stránek i tlačítko „logout“. Toto tlačítko slouží k odhlášení ze systému.

## 4.5. Struktura webového adresáře

Soubory tvořící systém jsou rozděleny do několika složek podle funkce, kterou vykonávají.

Složka **administrace** obsahuje skripty php, na které se odkazujeme při zadávání nových značek brýlových obrub, skel a kontaktních čoček do databáze. Tyto typy jsou posléze využity v seznamech, uvnitř formulářových oken.



Ve složce **css** je uložen soubor, který slouží k formátování generovaných webových stránek.

Složka **inc** je tvořena skripty ve formátu php. Jedná se o soubory načítané do dalších skriptů klauzulí include\_once(). Jde o konfigurační soubor obsahující funkci ověřující uživatelské přihlášení a funkci connect(), která zajišťuje spojení s databází.

Uvnitř složky **js** je uložen soubor functions.js, ve kterém jsou obsaženy funkce pro tvorbu datumu v jednotném formátu a výše zmíněná funkce kalkulačky plateb. Skript je naprogramován v java-skriptu.

Poslední složka **pic** je místem, kde jsou uloženy obrázky využívané v systému. Jde o logo a ikonky akcí.

Ostatní php skripty jsou uloženy přímo v kořenu webového serveru, jak naznačuje obrázek.

## 4.6. Způsob generování stránek a obsahu

Vůbec nejdůležitějším skriptem celého systému je soubor `template.php`. Tento soubor je načítán každým dalším skriptem. Prostřednictvím funkce `html()` v něm obsažené, jsou dynamicky generovány veškeré stránky.

**Funkce `html($title, $content, $left = array())`** je tvořena třemi parametry:

### **\$title:**

Tento parametr určuje, jakým názvem bude generovaná stránka označena. Hodnota tohoto parametru je předávána ze skriptu `value.php`, který bude popsán dále. Obsahem je tedy text, například „Nový zákazník“ nebo „Vyhledané osoby“.

### **\$content:**

Hodnotou dalšího parametru předávaného ze skriptů je zdrojový kód generovaného obsahu, v jazyce HTML. Jedná se například o kód pro tvorbu nové zakázky nebo prohlížení zakázky staré.

### **\$left:**

Jedná se o parametr typu pole. V tomto poli je uložen dynamicky generovaný obsah, který tvoří levou část stránky.

### **Struktura skriptu `template.php`**

Strukturu by měl nejlépe popsát komentář, který je součástí kódu.

```
<?php
// skript, který slouží k připojení k databázi a obsahuje nastavení
include('config.php');
// připojení k databázi (fce je součástí config.php)
connect();
// funkce generující obsah stránky
function html ($title, $description, $content, $left = array()) {
// v proměnné $left_str je uložen obsah levé části stránky
    $left_str = '';
    for ($lefti=0; $lefti<count($left);$lefti++){
        $left_str .= $left[$lefti];
    }
// tento kód vkládá do hlavičky jméno pobočky a zaměstnance
    if ($_SESSION["user"] == true){
        $sql = mysql_query('SELECT u.name AS uname,b.name AS bname
                            FROM users AS u
                            LEFT JOIN branch AS b ON (u.id_branch = b.id)
                            WHERE u.id = '.$_SESSION["user"]);
        $alias = mysql_fetch_array($sql);
        $head = '<span>Pobočka: '.$alias["bname"].'</span><br />
        <span>'. $alias["uname"].'<a href="/logout.php">logout</a></span>';
    }
// Proměnná $result nese html kód generované stránky
    $result = '';
    $result .= '<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
```

```

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="content-type" content="text/html; charset=utf-8"
  />
    <meta http-equiv="content-language" content="cs" />
    <meta name="resource-type" content="document" />
    <meta name="description" content="" />
    <meta name="keywords" content="" />
    <meta name="author" content="Lukáš Dlouhý" />
    <!-- Načtení javaskriptu -->
    <script type="text/javascript" src="js/functions.js"></script>
    <!-- Vložení kaskádových stylů -->
    <link href="../../css/styles.css" media="screen" rel="stylesheet"
type="text/css" />
    <title>Optika</title>
  </head>
  <body>
    <!-- Rozvržení stránky -->
    <div id="page">
      <!-- Záhlaví -->
      <div id="header">
        '.$head.'
      </div>
      <!-- Středová část okna stránek -->
      <div id="main">
        <!-- Obsah levé části -->
        <div id="left">
          '.$left_str.'
        </div>
        <!-- Obsah pravé části -->
        <div id="right">
          <!-- Nadpis okna -->
          <h1>'.$title.'</h1>
          <!-- Hlavní obsah -->
          <div id="content">
            '.$content.'
          </div>
        </div>
        <!-- Div, který vyčistí obtékání divu left a divu right -->
        <div class="cleaner"></div>
      </div>
      <!-- Patička -->
      <div id="footer"></div>
    </div>
  </body>
</html>';
// příkaz vrací výsledek funkce
return $result;
}
?>

```

## Skript object.php

Systém se skládá z mnoha formulářů. Jak bude okno s formulářem vypadat, které kolonky v něm budou a jaké je jeho rozvržení, definuje skript object.php a také soubor kaskádových stylů. V tomto souboru jsou všechny formuláře definovány pro vybranou akci se zvoleným typem zboží. Obsah souboru je tedy členěn na bloky dle vybrané akce a ty jsou dále zpracovány a předány jako obsah proměnné \$content skriptu template.php. Skript se také stará o naplnění kolonek daty, v případě, že chce uživatel informace pouze prohlížet.

## Zjednodušená struktura skriptu object.php

Tento skript je, co se týče délky obsahu, vůbec nejdelším a přesahuje i rozsah bakalářské práce. Z tohoto důvodu zde poskytnu pouze popis jeho struktury, nikoli části zdrojového kódu.

```
<?php
include('inc/template.php');
secure();
/* 1. Implicitní nastavení parametru akce na tvorbu nové zakázky
   (prodej brýlí) */
/* Levé menu */
/* 2. generování levého menu pro tvorbu nových akcí, odesílá se jako
   hodnota pole proměnné $left1 */
/* 3. výpis všech minulých činností zákazníka, také se odesílá jako
   hodnota pole proměnné $left2 */

/* Šablony pro generování formulářových oken a pro výpisy */
/* 4. šablona formuláře pro generování prodeje brýlí */
/* 5. šablona formuláře pro generování editace prodeje brýlí */
/* 6. šablona formuláře pro generování kontroly brýlí */
// další šablony...
?>
```

Z každého bloku kódu jsou odesílána data prostřednictvím již zmiňované funkce html(). Tato funkce je načtena na začátku, ze skriptu template.php. Část, kterou skript zpracuje, je podmíněna výběrem akce z levého menu.

## 4.7. Formuláře

Typickým vstupem dat do systému je například stránka sloužící k zadávání nové zakázky (viz. Obr. 6).

**MLB SYSTEM**

Lukáš Dlouhý [logout](#) Pobočka: Vaňkovka

**Lukáš Dlouhý**

Akce	Předmět
prodej	Brýle
prodej	Brýle
prodej	Brýle

### Prodej Brýlí - nová zakázka

Datum: 13. 5. 200

P Sph Cyl Ax Prism Baze Add Vizus PD  
1 -0,5 90 0 0 0 1,25 32

L 1,5 0 0 0 0 0 1,25 32

Výška 18 Vzdálenost ☐ B ☒ D

Obruba Police 1528 531

Typ čoček Stylis 1,67 Transition

Úpravy Antireflex Crizal Alizé

Barva hnědá

Poznámka

**Platba**

Obruba	3500	Celkem	14150
Levá čočka	6500	Záloha	10000
Pravá čočka	5500	Recept	1000
Sleva brýle	10	Kartou	<input checked="" type="checkbox"/>
		Sleva čočky	0
Zůstatek	4150		

Obr. 6 Formulářové okno

Tento obrázek demonstruje konečné rozložení částí typické stránky s formulářem. V horní části se nachází vygenerovaná hlavička, kterou tvoří logo systému, jméno přihlášeného uživatele, jeho příslušnost k pobočce a tlačítko sloužící k odhlášení. Levou stranu okna tvoří blok, v němž je zobrazeno jméno zákazníka a odkazy na jeho předchozí zakázky. Tyto odkazy jsou tvořeny typem akce, kterou zákazník uskutečnil, dále předmětem jehož se týká a datem uzavření zakázky. Vypsání informací slouží jako rychlý přehled zakázek a prostřednictvím odkazů lze ihned zkontrolovat, jakých hodnot je třeba k tvorbě nové zakázky nebo ke kontrole produktu.



Hlavní část okna tvoří formulář, v tomto případě slouží k tvorbě nové zakázky dioptrických brýlí. Zajímavostí na této stránce je funkce java-skriptové kalkulačky v sekci platba. Kromě poznámky, která je společná všem zakázkám, jsou veškeré údaje ukládány jako další hodnoty do tabulky values.

## **5. Přínosy navrhovaného řešení**

### **5.1. Rychlost navrženého systému**

Rychlost navrženého systému záleží především na rychlosti webového serveru, protože jsou skripty php zpracovávány na jeho straně a ke klientovi se dostane až kód v jazyce html. Nicméně dotazy kladené na databázi probíhají s obrovskou rychlostí, jedná se o časy v jednotkách, případně desetinách milisekund. Delší dobu trvá zpracování programů v jazyce php.

Největší zbrzdění může znamenat pomalé internetové připojení. Protože se jedná o systém, ve kterém by se mělo často pracovat, navrhl jsem strohou grafiku a využil především formátování prostřednictvím kaskádových stylů, místo objemných obrázků, které by se stahovaly delší dobu.

### **5.2. Urychlení práce**

Při návrhu struktury jsem se snažil zkrátit dobu od přihlášení po zadávání dat a samotné odhlášení na minimum. Posoudit snadnost ovládání a jeho rychlost, mohou až uživatelé systému, to ukáže čas. Mezi rychlostí přechodu mezi jednotlivými internetovými stránkami a mezi okny podobné databázové aplikace pracující na počítači je podstatný rozdíl, který se však s rychlostí sítí a serveru minimalizuje. Abych rozdíl snížil, snažil jsem se eliminovat počet průchodů stránkami a počet kliků na nejnížší možnou úroveň.

### **5.3. Rozšiřitelnost systému**

Systém byl navržen tak, aby byl snadno rozšiřitelný o další akce. Důkazem toho že, lze systém velice snadno rozšířit je původně neplánovaná, ale nakonec naprogramovaná část, pro tvorbu reklamačních formulářů. K vytvoření této sekce stačilo pouze zadat novou položku do číselníku akcí, vložit požadované názvy do tabulek a provázat je. Jedná se pouze o několik položek a z časového hlediska je přidávání do databáze zanedbatelné.

Posledním krokem bylo vytvoření poměrně krátké části kódu, velice podobné předchozím, do souboru `object.php`, ve kterém jsou jednotlivé bloky uloženy a podmínkami vybírány. Tato část kódu je vytvořena v html a zbývající část tvoří dotaz v jazyce SQL.

Pokud by měla společnost zájem, bylo by možné vytvořit administrační systém, který by přidávání nových akcí na stejném základě umožňoval. Při současném rozsahu, ani při výhledu do blízké budoucnosti, nepovažuji tvorbu takového systému za opodstatněnou.

## **5.4. Přenositelnost**

Systémy založené na open source projektech, lze obvykle velice snadno implementovat na širokém spektru operačních systémů. Z tohoto důvodu jsem zvolil osvědčenou kombinaci; server Apache, databázi MySQL a programovací jazyk PHP. Díky tomu lze tento databázový systém spravovat nejen na operačních systémech společnosti Microsoft, ale například i na systémech založených na linuxovém jádře. Právě s těmito servery se na internetu setkáváme nejčastěji.

## **5.5. Nákladová stránka řešení**

Cenu, za kterou by se dal takový systém pořídit, odhaduji v řádech několika desítek tisíc korun. Návratnost investice může být zhruba za 2-3 měsíce, během kterých se uspoří investované prostředky na nákladech za telefonování. Rychlost návratnosti mohou ovlivnit i další nekvantifikované přínosy, viz níže.

Společnost je již dostatečně vybavena, jak výpočetní technikou, tak i programovým vybavením na to, aby mohla systém začít využívat. Internetové připojení je k dispozici taktéž a jeho rychlost je více než postačující. Z tohoto pohledu další náklady neočekávám.

Systém však vyžaduje webové sídlo s dostatečným technickým a především bezpečnostním zázemím, nejlépe s podporou ssl. Firma vlastní server na němž běží její webové stránky. V případě potřeby, lze hosting splňující požadavky systému, pořídit za necelé 2000 korun měsíčně, od některého z českých poskytovatelů.

## **5.6. Kvantifikované přínosy**

Společnost vynakládá na telefonování z důvodu zjišťování potřebných informací o zákazníkovi, přibližně 2-3 tisíce korun na pobočku měsíčně. To je celkem 10-15 tisíc korun za měsíc na všechny pobočky. Roční náklady v současnosti přesáhnou stotisícovou částku. Tyto náklady budou se zavedením systému zcela odstraněny.

## **5.7. Nekvantifikované přínosy**

Zaměstnancům budou k dispozici rychle dostupné informace v přehledné formě a z časového hlediska aktuální. Údaje o zákaznících budou moci sdílet všechny pobočky.

Protože odpadne čekání zákazníků na zjištění naměřených hodnot z vyšetřovacích karet, z jiných poboček, lze očekávat udržení renomé společnosti, případně jeho zvýšení. Lze předpokládat, že klienti budou nakupovat právě v těchto pobočkách a klientela, která bude se službami spokojena, může dále rozšířit počet zákazníků.

Ze systému je možné získávat i statistiky o prodeji jednotlivých značek brýlí, kontaktních čoček a dalšího sortimentu firmy. Těmito výstupy se můžou vedoucí pracovníci řídit při nákupu zboží a udělat si přehled o klientele, která chodí do jednotlivých prodejen nakupovat nejčastěji. Na základě získaných informací lze přizpůsobovat i propagační akce, které ve svém důsledku zvedají tržby.

## 6. Závěr

Při zpracovávání bakalářské práce jsem oprášil a získal nové vědomosti, potřebné pro tvorbu informačních databázových systémů.

Samotná práce je rozdělena do několika kapitol. První kapitola „Analýza problému a současné situace“, je vytvořena na základě informací, které jsem zjistil při konzultacích se zaměstnanci společnosti a jejím vedením. Jedná se o informace týkající se komunikace pracovníků, jejich potřeb a historie společnosti.

Pro zpracování kapitoly „Teoretická východiska práce“ bylo nutné důkladně se zabírat vlastním řešením problému. Vzhledem k tomu, že jsem již měl nějaké zkušenosti s podobnými projekty, věděl jsem, kde potřebné informace hledat. Na tomto základě jsem začal projektovat daný systém.

Stěžejní částí bylo vytvoření funkčního systému pro kapitolu „Vlastní návrh řešení“. V této kapitole, ve značně zkrácené a zhuštěné podobě, popisuji vytvořený databázový systém. Bohužel mi bakalářská práce neumožňuje detailní rozpracování popisu celého kódu a důvodů, proč jsem systém takto zpracoval, v celém jeho rozsahu. Okomentoval jsem tedy jeho nejdůležitější vlastnosti a počítám s důkladnějším popisem při jeho obhajobě.

Kapitolou, která slouží ke kontrole, zda byly odstraněny zjištěné nedostatky, nese název „Přínosy navrhovaného řešení“. Právě zde se zabývám efektivitou systému a jeho kvantifikovanými a nekvantifikovanými přínosy pro společnost.

Databázový systém vytvořený v této práci, splňuje vytyčené cíle práce. Je jím zajištěno především získávání aktuálních dat ze všech poboček, avšak záleží také na tom, zda do něj budou pracovníci společnosti důsledně zadávat informace a včas provádět jejich změny.

Z hlediska ovladatelnosti systému, bylo dosaženo minimalizace kroků, které jsou třeba k tvorbě jednotlivých akcí. Toto přizpůsobení dělá systém snadno ovladatelný a urychluje a zpříjemňuje práci s ním.

Vzhledem k tomu, že se ve světě bitů a bajtů otázky bezpečnosti mění ze dne na den, nemohu zaručit plné zabezpečení systému. Pokud bude systém prověřen a opraven

nějakým bezpečnostním technikem a na straně serveru bude pracovat svědomitý správce serveru, měla by se stát aplikace vhodnou k dennímu využití.

Pokud se podívám na systém ze stránky designu, hodnotím ho jako strohý přiměřeně k nárokům na vytěžování datové sítě a jako vhodný pro jeho každodenní použití. Díky jeho návrhu a využitým technologiím je změna designu snadná a rychlá, stejně jako přemístění vstupů ve formulářích.

Věřím, že systém bude společnosti k užitku a doufám, že informace nabyté při zpracování bakalářské práce, využiji při tvorbě dalších, podobně zaměřených projektů.

## Seznam použitých zdrojů

### Seznam internetových zdrojů

Webové sídlo společnosti Netcraft Ltd. - [www.netcraft.com](http://www.netcraft.com)

Organizace W3C - [www.w3.org](http://www.w3.org)

Otevřená encyklopedie Wikipedie - <http://cs.wikipedia.org>

PC Svět - [www.pcsvet.cz](http://www.pcsvet.cz)

Sídlo komunity Open Source - <http://sourceforge.net>

Portál „Jak psát web“ - [www.jakpsatweb.cz](http://www.jakpsatweb.cz)

### Seznam tištěné literatury

ULLMAN, L.: *PHP a MySQL, Národní průvodce tvorbou dynamických WWW stránek.* Brno, Computer Press 2004, 534 s. ISBN 80-251-0063-4

POKORNÝ, J., HALAŠKA, I.: *Databázové systémy.* Praha, Česká infromatická společnost 2002, 146 s. ISBN 80-900853-9-3

HLAVENKA, J. et al.: *Vytváříme WWW stránky a spravujeme moderní web site, 4. aktualizované vydání.* Praha, Computer Press 2000, 520 s. ISBN 80-7226-293-9

NARAMORE, E. et al.: *Vytváříme webové aplikace v PHP5, MySQL a Apache.* Brno, Computer Press 2006, 813 s. ISBN 80-251-1073-7

BRÁZA, J.: *PHP 5, začínáme programovat.* Praha, Grada 2005, 244 s. ISBN 80-247-1146-X

CASTAGNETTO, J. et al.: *Programujeme PHP profesionálně.* Brno, Computer Press 2004, 656 s. ISBN 80-7226-310-2

## Seznam obrázků

Obr. 1 Hierarchie ve společnosti.....	12
Obr. 2 Formátování CSS .....	24
Obr. 3 Třívrstvá architektura projektu.....	27
Obr. 4 Entitně-relační diagram .....	31
Obr. 5 Kalkulátor plateb .....	35
Obr. 6 Formulářové okno .....	40

## Seznam příloh

Příloha 1 – stránka „Rozcestník“ .....	49
Příloha 2 – stránka vyhledaných osob .....	49
Příloha 3 – editační rozhraní .....	50



## Příloha 1 – stránka „Rozcestník“



Lukáš Dlouhý [logout](#) Pobočka: Vaňkovka

Vyhledat zákazníka

**Rozcestník**

## Příloha 2 – stránka vyhledaných osob



Lukáš Dlouhý [logout](#) Pobočka: Vaňkovka

• [Rozcestník](#)

Vyhledat zákazníka

**Vyhledané osoby**

☒ **Big Samson** (Válečníků 1, Trója)

☐ **Černý Jakub** (Na Palcích, Brno)

☐ **Kompanová Barbora** (Mučednická 33, Brno)


**Vyber akci.**

prodej

**Vyber druh zboží.**

Brýle

### Příloha 3 – editační rozhraní



Lukáš Dlouhý

logout

Pobočka:Vaňkovka

Nová akce pro toho  
zákazníka:

Vyber akci.

- Zvolte akci -

Vyber druh zboží.

- Zvolte předmět -

přejít

Lukáš Dlouhý

Akce	Předmět
prodej	Brýle
prodej	Brýle
prodej	Brýle
prodej	Brýle

Editace prodeje brýlí

Datum: 13. 5. 2007

	Sph	Cyl	Ax	Prism	Baze	Add	Vizus	PD
P	1	-0,5	90	0	0	0	1,25	32
L	1,5	0	0	0	0	0	1,25	32

Výška 18

Vzdálenost ☐ B ☒ D

Obruba Police 1528 531

Typ čoček Stylis 1,6 Úpravy

Antireflex Crizal Alizé

Barva hnědá

Poznámka

ulož

Platba

Obruba	3500	Celkem	14150
Levá čočka	6500	Záloha	10000
Pravá čočka	5500	Recept	1000
Sleva brýle	10	Kartou	<input checked="" type="checkbox"/>
		Sleva čočky	0
Zůstatek	4150		

50